AMENDMENTS TO THE CLAIMS:

1. (*Currently Amended*) A method of constructing route information for a multi-gigabit switching router comprising ~~the steps of~~:

partitioning each route prefix of route information into a segment and an offset, ~~and~~ said offset having variable length;

building a segmentation table based on said segment, said segmentation table comprising a pointer/next hop field and an offset length field;

constructing a Next Hop Array for each entry in said segmentation table when ~~the~~ <u>a</u> longest prefix length of said segment is larger than a first predetermined value;

constructing a Compression Bit Map for each entry of said Next Hop Array when ~~the~~ <u>a</u> value of said offset length field is larger than a second predetermined number of bits; and

constructing a Code Word Array and a Compressed Next Hop Array for each ~~of~~ said Compression Bit Map, said Code Word Array ~~consisted~~ <u>consisting</u> of a plurality of code words each having a Map and a Base.

2. (*Currently Amended*) The method as claimed in claim 1, further comprising ~~the steps of~~:

using a first bit stream of an IP address of an incoming IP packet as an index to look up said segmentation table;

determining ~~if~~ <u>whether</u> said first bit stream of an entry of said segmentation table pointed to by said index is larger than or equal to M;

outputting said entry of said segmentation table as a next hop when said first bit stream of said entry is smaller than said M~~.~~<u>;</u>

determining a correspondent entry of said segmentation table is a pointer pointing to a corresponding Next Hop Array when said first bit stream of said entry of said segmentation table is larger than or equal to said M, and a second bit stream of said corresponding entry of said segmentation table is smaller than or equal to Y, and using said second bit stream of said corresponding entry of

said segmentation table as an index to look up said corresponding Next Hop Array.

3. (Original) The method as claimed in claim 2, wherein said first bit stream is the leftmost 20 bits, said second bit stream is the rightmost 4 bits, said M is 256, and said Y is 3.

4. (*Currently Amended*) The method as claimed in claim ~~1~~ 2, further comprising ~~the steps of~~:

determining a correspondent entry of said segmentation table is a pointer pointing to a corresponding Code Word Array when a first bit stream of said entry is larger than or equal to M, and a second bit stream of said correspondent entry of said segmentation table is larger than said Y;

computing an index for looking up a corresponding code word in said Code Word Array by adding said correspondent entry of said segmentation table, which is a pointer, plus said second bit stream;

computing an index for looking up a corresponding Compressed Next Hop Array by adding (said pointer + $2^{k-4} \times 4 - 1$), a Base of said corresponding Code Word and $|w|$, said $|w|$ representing the number of "1"s accumulated from the *0-th* bit to *w-th* bit of the map of the code word of said corresponding Code Word Array, ~~and~~ said k representing an offset length.

5. (Original) The method as claimed in claim 1, wherein said pointer/next hop field of said segmentation table is 20 bits and said offset length field is 4 bits.

6. (Previously Presented) The method as claimed in claim 1, wherein each of said Next Hop Arrays contains 2^k entries, and said k is determined by the longest prefix length of each segment.

7. (Original) The method as claimed in claim 1, wherein said step of constructing a Compressed Bit Map and said step of constructing a Compressed Next Hop Array can be executed at the same time.

8. (*Currently Amended*) The method as claimed in claim 7, ~~comprises~~ comprising ~~the steps of~~:

reading a set of sorted route prefixes of a segment in an increasing order by the length of prefixes, and each pair of start point and end point of said list of sorted route prefixes is sorted according to ~~the~~ an order in said set of route prefixes of said segment ;

sorting each element in said set in an order according to its memory address in said segment;

processing each element in said set from left to right and in a manner that:

(a) determining if a selected element is a start point S_i^0, where "i" represents the *i-th* route prefix in said set and "0" represents ~~the~~ a number of update times of said start point in the memory; when said selected element is a start point S_i^0, executing step (b), and when said selected element is not a start point S_i^0, executing step (c);

(b) pushing said start point S_i^0 onto a stack, and appending said start point S_i^0 to an array; ~~repeate~~ and repeating said step (a) until each element in said set is processed;

(c) removing a top element of said stack;

(d) determining if the top element of said stack is a start point S_j^k, where "*j*" represents the *j-th* route prefix in said set and "k" represents the number of update times of said start point in the memory; when said top element is said start point S_j^k, executing step (e), and when said top element is not said start point S_j^k, executing step (f);

(e) appending S_j^{k+1} to said array where the next hop of a start point S_j^{k+1} is equal to the next hop of a start point S_j^k, and ~~the~~ a memory address of a start point S_j^{k+1} is equal to ~~the~~ a memory address of an end

address E_i^0+1; and replacing the top element of said stack with said start point S_j^{t+1}; ~~repeate~~ and repeating said step (a) until each element in said set is processed;

(f) executing nothing; ~~repeate~~ and repeating said ~~setp~~ step (a) until each element in said set is processed;

compacting said array in a manner that for any consecutive element S_j^t and S_p^q, removing S_j^t from said array if the memory address of said start point S_j^t is equal to the memory address of said start point S_p^q, and removing said start point S_p^q from said array if the next hop of said start point S_j^t is equal to said next hop of said start point S_p^q;

removing each element S_j^t from said array when said memory address of said start point S_j^t is equal to the memory address of the end point E_0^0; and

concurrently constructing a Compression Bit Map Array and a Compressed Next Hop Array by assigning "1" for each start point in said array to generate said Compression Bit Map Array and assigning ~~an~~ a next hop of said start point in said array to each corresponding entry of the Compressed Next Hop Array to generate said Comrpessed Next Hop Array.

9. (Original) The method as claimed in claim 1, wherein said first predetermined value is 16 and said second predetermined number of bits is 3.

10. (Original) An IP routing lookup system for a multi-gigabit switching router comprising:

means for partitioning an incoming IP address into a segment portion and an offset of variable length portion;

means for storing a segmentation table, said segmentation table comprising a pointer/next hop field and an offset length field, and said segmentation table storing means coupled to said partitioning means for looking up an entry in said segmentation table using said segment portion as an index;

means for storing a plurality of Next Hop Arrays and capable of

outputting a next hop in response to an index;

means for storing a plurality of Code Word Arrays, each Code Word Array consisting of a plurality of code words and each code word comprising a Map and a Base for indexing a Compressed Next Hop Array;

first adding means for adding a plurality of mask bits of Map bits of a corresponding code word and generating the value of $|w|$, said $|w|$ representing the number of "1"s accumulated from the *0-th* to *w-th* bit of said code word of said corresponding Code Word Array;

second adding means for adding a pointer read from the pointer/next hop field of said segmentation table plus a parameter, said base, and said $|w|$ to generate a second index pointing to said Compressed Next Hop Array;

means for generating a next hop from said Compressed Next Hop Array in response to said second index;

means for determining (1) whether the first bit stream of an entry of said segmentation table is larger than or equal to M, (2) whether said first bit stream of an entry of said segmentation table is larger than or equal to said M and the second bit stream of an entry of said segmentation table is larger than Y, and (3) whether the first bit stream of an entry of said segmentation table is larger than or equal to said M and the second bit stream of an entry of said segmentation table is smaller than or equal to Y; and

means for selecting an output from said segmentation table, said Next Hop Array, and said Compressed Next Hop Array in response to a determination of said determining means, whereby in said condition (1) selecting output from said segmentation table, in said condition (2) selecting output from a corresponding Compressed Next Hop Array, and in said condition (3) selecting output from a corresponding Next Hop Array.

11. (*Currently Amended*) The system as claimed in claim 10, wherein said segment portion is 16 bits and said offset of variable length portion is less <u>than</u> or equal to 16 bits.

12. (Original) The system as claimed in claim 10, wherein said first bit stream is the leftmost 20 bits, said second bit stream is the rightmost 4 bits, said M is 256, and said Y is 3.

13. (Original) The system as claimed in claim 10, wherein said pointer/next hop field of said segmentation table is 20 bits and said offset length field is 4 bits.

14. (*Currently Amended*) The system as claimed in claim 10, wherein each of said Next Hop Arrays contains 2^k entries, and k is determined by ~~the~~ a longest prefix length of each segment.

15. (Original) The system as claimed in claim 10, wherein said Map is 16-bit and said Base is 16-bit.

16. (Original) The system as claimed in claim 10, wherein each entry of said Compressed Next Hop Array is 8-bit, and each entry of said Next Hop Array is 8-bit.

17. (*Currently Amended*) ~~The~~ A method of constructing route information for a multi-gigabit switching router comprising ~~the steps of~~:

partitioning each route prefix of route information into a segment and an offset, ~~and~~ said offset having a variable length;

building a segmentation table based on said segment, said segmentation table comprising a pointer/next hop field and an offset length field;

constructing a Next Hop Array for each entry in said segmentation table when ~~the~~ a longest prefix length of said segment is larger than a first predetermined value;

constructing a Compression Bit Map for each entry of said Next Hop Array when ~~the~~ a value of said offset length field is larger than a second

predetermined number of bits;

constructing a Code Word Array and a Compressed Next Hop Array for each of said Compression Bit Map, said Code Word Array ~~consisted~~ consisting of a plurality of code words each having a Map and a Base;

using a first bit stream of an IP address of an incoming IP packet as an index to look up said segmentation table;

determining ~~if~~ whether said first bit stream of an entry of said segmentation table pointed to by said index is larger than or equal to M;

outputting said entry of said segmentation table as a next hop when said first bit stream of said entry is smaller than said M;

determining said correspondent entry of said segmentation table is a pointer pointing to a corresponding Next Hop Array when said first bit stream of said entry of said segmentation table is larger than or equal to said M, and said second bit stream of said corresponding entry of said segmentation table is smaller than or equal to Y, and using said second bit stream of said corresponding entry of said segmentation table as an index to look up said corresponding Next Hop Array;

determining said correspondent entry of said segmentation table is a pointer pointing to a corresponding Code Word Array when said first bit stream of said entry is larger than or equal to said M, and said second bit stream of said correspondent entry of said segmentation table is larger than said Y;

computing an index for looking up a corresponding code word in said Code Word Array by adding said correspondent entry of said segmentation table, which is a pointer, plus said second bit stream; and

computing an index for looking up a corresponding Compressed Next Hop Array by adding (said pointer + $2^{k-4} \times 4 - 1$), a Base of said corresponding Code Word and $|w|$, said $|w|$ representing the number of "1"s accumulated from the *0-th* to *w-th* bit of said corresponding Code Word Array, and said k representing an offset length.

18. (Original) The method as claimed in claim 17, wherein said first predetermined value is 16, said second predetermined number of bits is 3, said first bit stream is the leftmost 20 bits, said M is 256, said second bit stream is the rightmost 4 bits, and said Y is 3.

19. (*Currently Amended*) The method as claimed in claim 17, wherein said pointer/next hop field of said segmentation table is 20 bits and said offset length field is 4 bits.

20. (Previously Presented) The method as claimed in claim 17, wherein each of said Next Hop Arrays contains 2^k entries, where k is determined by the longest prefix length of each segment.